AF/2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Ehnebuske et al.**

Serial No.: **09/204,971**

Filed: **December 3, 1998**

For: **Apparatus and Method for Performing General Integrity Checks Using Integrity Rule Checking Points in an Enterprise Application**

§ Group Art Unit: **2122**
§
§ Examiner: **Ingberg, Todd D.**
§
§ Attorney Docket No.: **AT9-98-267**
§

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on April 22, 2002.

By: Dell Whitton
Dell Whitton

TRANSMITTAL DOCUMENT

Assistant Commissioner of Patents
Washington, D.C. 20231

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of $320.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Deposit Account No. 09-0447.

Respectfully submitted,

Duke W. Yee
Registration No. 34,285
CARSTENS, YEE & CAHOON, LLP
P.O. Box 802334
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANTS

Docket No. AT9-98-267 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Ehnebuske et al.** §
§ Group Art Unit: **2122**
Serial No. **09/204,971** §
§ Examiner: **Todd Ingberg**
Filed: **December 3, 1998** §
§
For: **Apparatus and Method for Performing General Integrity Checks Using Integrity Rule Checking Points in an Enterprise Application** §
§

Assistant Commissioner for Patents
Washington, D.C. 20231

ATTENTION: Board of Patent Appeals and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on April 22, 2002.

By: Dell Whitton
Dell Whitton

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 22, 2002.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines, Inc.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interference's that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-29

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-29
4. Claims allowed: NONE
5. Claims rejected: 1-29

C. CLAIMS ON APPEAL

The claims on appeal are: 1-29

STATUS OF AMENDMENTS

No amendments to the claims after final rejection have been made.

SUMMARY OF INVENTION

The present invention provides a method, apparatus and computer program product for performing general integrity checks using rules in an application running on a data processing system. A point is identified at which a unit of work is to complete. The unit of work includes a plurality of participants. Responsive to determining that the unit of work is to complete, rules associated with each participant in the unit of work are obtained. Responsive to obtaining the rules, the rules obtained for each of the participants are run. Responsive to running the rules, the general integrity of the system with respect to the unit of work is determined. Responsive to determining the general integrity of the application state, the unit of work is completed by committing it or aborting it.

ISSUES

The only issue on appeal is whether claims 1-29 are obvious in view of Martin, Principles of Object-Oriented Analysis and Design, 1993, in view of Code Complete and the IBM Computer Dictionary.

GROUPING OF CLAIMS

The claims do not stand or fall together for the reasons set forth hereafter in Appellants' arguments. The claims stand or fall according to the following grouping of claims:

Group I: claims 1, 7, 10, 12, 19, 24 and 25;
Group II: claims 8, 22 and 28;
Group III: claims 2, 3, 9, 20, 21, 23, 26, 27 and 29;
Group IV: claims 4 and 16;
Group V: claims 5 and 17;
Group VI: claims 6 and 18;
Group VII: claim 11;
Group VIII: claim 13;

Group IX: claim 14; and

Group X: claim 15.

ARGUMENT

I. Information Disclosure Statement and Definition of Terms

The issues regarding the filing of an Information Disclosure Statement and Definitions of Terms raised in the First Office Action have been withdrawn by the Examiner. However, Appellants, in view of the Examiner's statements in the first twenty pages of the Final Office Action with regard to these issues, wish to again reiterate that Appellants are under no obligation to research alleged prior art raised by the Examiner nor supply information regarding prior art raised by the Examiner of which Appellants are not aware of any materiality with regard to the patentability of the present invention, as defined by the pending claims. If the Examiner is aware of alleged prior art that the Examiner believes is material to patentability, the Examiner must cite the art on a PTO Form 1449 and cannot shift the burden to Appellants simply by raising the issue in an Office Action.

Furthermore, with regard to the definitions of terms asserted by the Examiner, again Appellants wish to reiterate that the terms in the pending claims are not limited by the Examiner's asserted definitions. Appellants have offered other example definitions of these terms as taken from the present specification. These definitions are not meant to be limiting with regard to interpretation of the claims but are only offered as examples. Similarly, any alleged agreement by Appellants with the Examiner's asserted definitions is only an agreement that the Examiner's definitions are adequate examples of how these terms may be interpreted in view of the specification. Such agreements and definitions are not "findings of fact" and are not intended to imply any limitations with regard to interpretation of the claims.

II. 35 U.S.C. § 103, Obviousness

The Final Office Action rejects claims 1-29 under 35 U.S.C. § 103(a) over Martin, Principles of Object-Oriented Analysis and Design, 1993 in view of alleged common knowledge

of programming as taught by Code Complete and definitions as provided by the IBM Computer Dictionary. The rejection is primarily based on Martin with Code Complete and IBM Computer Dictionary allegedly providing the teachings of using "common sense" in programming and some well-known terminology. This rejection is respectfully traversed.

With regard to claim 1, the Office Action states:

> **Martin** teaches a method for performing general integrity checks using rules in an application running on a data processing system (**Martin**, page 133, 138 – 142, structure of Rules) comprising: identifying a point in a unit of work where application state integrity is to be verified (**Martin**, page 143, object state rules), wherein the unit of work includes a plurality of participants (**Martin**, page 143, last paragraph, linked to appropriate items in OO diagram); obtaining rules associated with each participant in the unit of work (**Martin**, page 144, Box 10.3); responsive to obtaining the rules (**Martin**, page 136, Rules Linked to Diagrams), running the rule obtained for each of the participants to verify the integrity of an application state (**Martin**, page 143, object state rules), according to the plurality of participants; general integrity checks running on a data (**Martin**, page 133, 138 – 142, structure of Rules). Martin does not explicitly teach the programming constructs of responsive to determining that the unit of work is to be completed. Although, Martin clearly supports Rules and the testing for conditions the reference does not explicitly states RULES can be used to determine if a write operation should be performed or not. The following overviews the teaching of **Martin**.
>
> **Martin** Reference teaches RULES
>
> Testing the Integrity of an Object (page 143)
>
> When condition (Chapter 10 a format for Rules logic structure testing the state of the Object)
>
> Perform operation
>
> When condition
>
> Perform operation
>
> (**Note**: the construct closely resembles the structure of a CASE/switch statement in many languages)
>
> What **Martin** does give is the endless possibilities of the conditions and the endless possibilities of the operations, such as if a state condition is wrong don't write (negative results) if the state operation is correct (positive results) perform a write operation. Examiner, holds determinations when to perform a write operation and when not to perform a write are issues of normal use and considered

part of being a artisan of ordinary skill in the art. The term for employing common sense in programming is called Defensive Programming. It is the reference **"Code Complete"** by Steve McConnell that teaches defensive programming like on page 97 of **Code Complete** "Garbage In Does Not Mean Garbage Out". In other words test before you use data. Therefore, it would have been obvious to one of ordinary skill at the time of invention to use the teaching of Martin's RULES to perform defensive programming, because "...it's the recognition that programs will have problems and modifications, and that a smart programmer will develop code accordingly". **(Code Complete**, page 94, 5.6 Defensive Programming, Key Point).

Claim 1, which is representative of claims 19 and 25 with regard to similarly recited subject matter, reads as follows:

> 1. A method for performing general integrity checks using rules in an application running on a data processing system comprising:
>
> identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;
>
> responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and
>
> responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants.

It should first be noted that the Martin reference is a reference comprising over 400 pages of text. Therefore, Appellants are only addressing those sections explicitly cited by the Final Office Action. Although the Examiner asserts, in the Final Office Action, that the reference must be taken as a whole, it is the Examiner's burden to cite with specificity, the sections of the reference that the Examiner believes support a finding of obviousness. It is not Appellants' duty to delve through 400 pages of text to recreate the Examiner's reasoning. Such a requirement would shift the burden of establishing a prima facie case of obviousness from the Examiner and place it on Appellants. Thus, Appellants are addressing the teachings of Martin as explicitly cited by the Examiner. Moreover, even taking Martin as a whole, the reference does not teach any of the features of the present invention, as detailed hereafter.

Martin is a general textbook that includes a section on the use of rules in object-oriented programming. While Martin generally teaches the use of rules, and even integrity rules, Martin does not teach or even suggest the specific use of rules set forth in claims 1, 19 and 25. A

general teaching does not render every specific use of the general teaching obvious. The following are Appellants' responses to the Final Office Action's allegations regarding the specific features recited in claims 1, 19 and 25. Martin does not teach or suggest any of these features.

The Office Action alleges that Martin teaches a method of performing general integrity checks using an application running on a data processing system at pages 133 and 138-142. While it is true that Martin does teach the use of integrity rules and states that these rules indicate that something must be true (page 137), Martin does not teach the specific use of integrity rules set forth in claims 1, 19 and 25 (hereafter only referred to as claim 1), as discussed below.

The Office Action alleges that Martin teaches identifying a point in a unit of work where application state integrity is to be verified merely because Martin allegedly teaches object state rules at page 143. First, Martin does not teach or even suggest units of work as the feature is recited in the pending claims. A unit of work is a piece of business work which defines each business context in which it is carried out. While Martin teaches that business policies may be represented as rules (see page 133), Martin makes not mention or even suggestion regarding units of work. Furthermore, business policies are not the same as units of work. A business policy merely outlines the decision processes of a business system, it does not represent a unit of work that is to be performed.

Because Martin does not teach units of work, Martin cannot be found to teach or even suggest identifying a point in a unit of work where application state integrity is to be verified. While Martin teaches object state rules on page 143, all that is stated is "Object state rules are identified in object-structure analysis. They are associated with diagrams, such as the data-structure diagram, object-relationship diagram, or composed-of diagram." It is not seen how such a general statement somehow teaches the very specific feature of identifying a point in a unit of work where application state integrity is to be verified, as recited in claim 1. The description of object state rules on page 143 of Martin does not make any mention of units of work, let alone identifying a point in a unit of work where state integrity is to be verified.

Furthermore, even if it were interpreted that Martin teaches units of work, Martin still does not teach identifying a point in a unit of work where application state integrity is to be verified. As noted above, the general statement that there are object state rules, that they are identified in object-structure analysis, and that they may be associated with diagrams, has nothing

to do with identifying a point in a unit of work where application state integrity is to be verified. The Examiner is reading features into the Martin reference that are simply not there in order to arrive at Appellants' claimed invention having first had benefit of Appellants' disclosure. In other words, the Examiner is engaged in hindsight reconstruction in which the Examiner is conjuring teachings from the reference that simply are not there.

The Final Office Action alleges that Martin teaches that a unit of work includes a plurality of participants merely because Martin allegedly mentions "three more rule windows that are linked to appropriate items in OO diagrams" (page 143, last paragraph). It is not understood how linking rule windows to object oriented diagrams teaches a unit of work having a plurality of participants. There is no correlation between the feature recited in claim 1 and the cited portion of Martin. Linking rule windows to object oriented diagrams has nothing to do with a unit of work having a plurality of participants.

The Final Office Action further alleges that Martin teaches obtaining rules for each of the participants in the unit of work merely because Martin provides example of rules associated with diagrams of object oriented analysis in Box 10.3 on page 144. Box 10.3 of Martin does not teach a unit of work, let alone obtaining rules for each participant in the unit of work. The only thing that Box 10.3 teaches is examples of rules for different types of object oriented diagrams. There is no teaching or even suggestion in Box 10.3 of using units of work, and definitely no teaching or suggestion of obtaining rules for each participant in a unit of work. Again, the Examiner is engaging in hindsight reconstruction by reading in teachings to the Martin reference that simply are not there.

In addition, the Final Office Action alleges that Martin teaches running the rules obtained for each of the participants to verify the integrity of an application state merely because Martin allegedly teaches object state rules on page 143. The general teaching of object state rules does not provide any teaching or suggestion regarding running rules obtained for participants in a unit of work to verify the integrity of an application state. The general teaching of object state rules makes no teaching or suggestion of units of work, rules for participants in a unit of work, obtaining and running rules for participants in a unit of work, or that there is any correlation between rules for participants in a unit of work and an application state. There simply is no correspondence between the section of Martin cited by the Final Office Action and the features recited in claim 1.

In summary, Martin has nothing to do with the invention recited in claim 1. While Martin provides a number of general teachings, they do not teach or suggest any of the features recited in claim 1. Furthermore, the Examiner appears to be stretching the interpretation of Martin in an attempt to reach Appellants' claimed invention beyond any reasonable interpretation of the reference. Such a stretch of interpretation is based on hindsight reconstruction using Appellants' own disclosure as a guide.

Appellants do agree with the Final Office Action that Martin does not teach obtaining rules for each of the participants in a unit of work in response to determining that a unit of work is to be completed, as recited in claim 1. This is because Martin does not teach a unit of work, participants in a unit of work, or even that a unit of work may be completed. However, Appellants disagree with the Final Office Action regarding the allegation that Martin provides "endless possibilities" of conditions and operations such as write operations and that when to perform a write operation and when not to perform a write operation somehow makes obvious this feature. The Final Office Action also cites Code Complete as teaching defensive programming which the Final Office Action somehow links with Martin to allegedly modify the Martin reference so that a determination that a unit of work is to be completed is somehow made obvious.

First, the Appellants cannot follow the Examiner's reasoning. It is not at all clear how a general teaching of conditions can be found to teach the specific feature set forth in claim 1 of a determination that a unit of work is to be completed. The Final Office Action states that the conditions of when to write and when not to write make this feature obvious. Appellants do not claim to be the first to invent determining if a unit of work is to be completed. Rather, Appellants claim obtaining rules for participants in the unit of work when it is determined that a unit of work is to be completed. Thus, the Final Office Action's example of a write condition has no bearing on the presently claimed features.

Second, it is not clear how a general teaching of "defensive programming" may somehow be combined with the general teachings provided in Martin to arrive at the specific method set forth in claim 1. The Code Complete reference does not provide any teachings that cure the deficiencies in Martin set forth above. In other words, the general teaching of "defensive programming" does not render obvious "identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;

responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants," as recited in claim 1.

In addition, the IBM Computer Dictionary provides no additional teaching regarding any of the features of claim 1. That is, the combination of the IBM Computer Dictionary with Martin and Code Complete does teach or suggest any of the features in claim 1. None of these references have anything to do with the presently claimed invention recited in claim 1. The Examiner has taken extremely general teachings in a textbook and attempted to stretch these teachings beyond reasonable interpretation, and read additional teachings into the references that are not there, in an attempt to encompass the very specific method set forth in claim 1. Such a stretch of the teachings and reading in of additional teachings is not supported by the references and is based solely on hindsight reconstruction using Appellants' own disclosure as guide.

With regard to claims 8, 22 and 28, neither Martin, Code Complete, or the IBM Computer Dictionary teaches or suggests detecting a commit for a unit of work. As set forth above, a commit in this context is a process for determining whether to commit the state changes made by the unit of work to the participants in the unit of work. The Final Office Action alleges that the mere teaching of having conditional rules in Martin somehow is the same as detecting a commit for a unit of work. Appellants respectfully disagree. Yet again, the Examiner is reading in teachings to the Martin reference that are not there. The general teachings of conditional rules does not teach or suggest detecting a commit of a unit of work.

As previously noted above, Martin does not teach a unit of work. Therefore, Martin cannot teach detecting a commit of a unit of work. Furthermore, Martin does not teach a commit of a unit of work. A general teaching of a condition rule does not provide any teaching or suggestion of a determination of whether to commit the state changes made by a unit of work to the participants in the unit of work, i.e. a commit of a unit of work.

In addition, claims 8, 22 and 28 include features that are similar to some of the features in claims 1, 19 and 25. Therefore, Martin, Code Complete and the IBM Computer Dictionary do not provide any teaching or suggestion of these features in claims 8, 22 and 28, as discussed above. With regard to claim 12, as previously noted, none of the references teach or suggest a

unit of work, participants in a unit of work, or locating rules for the participants in response to activation of the unit of work to complete the unit of work.

In view of the above, Appellants respectfully submit that neither Martin, Code Complete, or the IBM Computer Dictionary, either alone or in combination, teach or suggest the features recited in claims 1, 8, 12, 19, 22, 25, 28. At least by virtue of their dependency on claims 1, 8, 12, 19, 22, 25 and 28, respectively, none of the references either alone or in combination teach or suggest the features set forth in dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. Accordingly, Appellants respectfully request withdrawal of the rejection of claims 1-29 under 25 U.S.C. § 103(a).

In addition, none of the references teach or suggest any of the specific features set forth in the dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. The Final Office Action provides specific rejections of each of the dependent claims based on the combination of references discussed above. Each of these rejections is based on the same flawed interpretation and reading in of teachings discussed above and therefore, are traversed for similar reasons as set forth above.

In addition, because the dependent claims build off of the features recited in their respective independent claims, the deficiencies of the cited references are likewise applicable to the additional features set forth in the dependent claims. Moreover, none of the references teach or suggest any of the specific features recited in the dependent claims.

For example, because none of the references teach or suggest a unit of work or running rules for participants in a unit of work, as recited in claim 1, none of the references can be found to teach "responsive to a negative result obtained by running the rules, aborting the unit of work," as recited in claim 2. Moreover, none of the references teach or suggest committing a unit of work when a positive result is obtained by running the rules for the participants in the unit of work, as recited in claim 3. The Final Office Action alleges that these features are taught by the IBM Computer Dictionary as an abnormal termination and a commit. Appellants disagree that the aborting of a unit of work is the same as an abnormal termination. However, even if it were the same, the IBM Computer Dictionary does not teach aborting a unit of work or committing a unit of work based on a result obtained by running rules for a plurality of participants in a unit of work in response to a determination that a unit of work is to be completed, at a point in the unit of work where application state integrity is to be checked. This same distinction applies to

claims 9, 20, 21, 23, 26, 27 and 29 which recite similar features in their respective claim terminology.

With regard to claim 4, the Final Office Action alleges that the features of each participant having an associated name and obtaining rules based on the name associated with the participant is taught by Martin in that Martin teaches linking rules to diagrams. This in no way has anything to do with obtaining rules for a plurality of participants in a unit of work based on the names associated with the participants. This same distinction applies to claim 16 which recites a similar feature in its respective claim terminology.

Regarding claim 5, the Final Office Action alleges that the features that each participant is an object and wherein the name associated with the object is the class name of a participating object is taught by Martin in box 10.3 as the "customer." Appellants are at a loss as to what the Examiner's reasoning is. While "customer" may be a class name, there is nothing in Martin, and especially box 10.3, that teaches participants in a unit of work being objects whose names are class names and wherein the names of the objects that are participating in the unit of work are used to obtain rules that are run to verify application state integrity. This same distinction applies to claim 17 which recites a similar feature in its respective claim terminology.

With regard to claim 6, the Final Office Action alleges that the features of a unit of work being associated with a type and each participant being associated with a name, and the step of obtaining rules is based on the name of the participant and the type of the unit of work is taught by Martin at pages 136, 138-140 and 144 because Martin allegedly teaches rules linked to diagrams. There is nothing in these sections, or any other sections, of Martin that teaches types of units of work or names of participants in a unit of work. Moreover, there is nothing in Martin that remotely even resembles obtaining rules for participants in a unit of work based on the name of the participant and the type of unit of work. This same distinction applies to claim 18 which recites a similar feature in its respective claim terminology.

Regarding claim 11, the Final Office Action alleges that the feature of each rule associated with a unit of work having available for use each participant within the unit of work is taught by Martin simply because Martin teaches structures of rules. However, a general teaching of a rule structure, as set forth in Martin does not make any teaching or suggestion as to a rule being associated with a unit of work and that the rule associated with the unit of work may be used with each participant within the unit of work.

Regarding claim 13, the Final Office Action alleges that the feature of activation of a unit of work control point being initiated by a commit instruction to the unit of work is taught by Martin on pages 140-142. There is nothing on pages 140-142 that even mentions a unit of work, let along a unit of work control point being activated by a commit instruction.

With regard to claim 14, the Final Office Action alleges that the feature of a control point identifying applicable rules for all of the participants in the unit of work is taught by Martin at pages 140-141 because Martin teaches the ability to have conditionals. Conditional rules have nothing to do with the feature of a control point identifying rules that are applicable to all of the participants in a unit of work. There is no correlation between the Examiner's "conditionals" and the features recited in claim 14.

Regarding claim 15, the Final Office Action alleges that the feature of a control point applying applicable rules to a portion of the participants in a unit of work is taught by Martin at pages 140-141 because Martin teaches the ability to have conditionals. Conditional rules have nothing to do with the feature of a control point identifying rules that are applicable to a portion of the participants in a unit of work. There is no correlation between the Examiner's "conditionals" and the features recited in claim 15.

Thus, in addition to being dependent on independent claims 1, 8, 12, 19, 22, 25, 28, the dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29 are also allowable over the cited references by virtue of their specific recited features. Accordingly, the rejection of claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29 should be withdrawn.

Examiner's Response to Above Arguments and Appellants' Rebuttal

In response to the above arguments, the Examiner, in the Final Office Action, alleges:

1) Martin provides teachings on Object Oriented CASE tools – Rules and common programming constructs (Final Office Action, page 39);

2) Appellants allegedly admit that the Martin reference teaches "units of work" (Final Office Action, page 40);

3) Martin teaches a Rule for integrity checking and rules may be attached to any diagram (as in any point) and mentions the diagrams are executable (Final Office Action, page 41);

4) Any judgment of obviousness is necessarily a reconstruction based on hindsight reasoning (Final Office Action, pages 41-42);

5) The Martin reference, when taken as a whole, teaches object oriented CASE tools for enterprise solutions and enterprises have many participants (categories of objects in Chapter 6 and interaction among objects in Chapter 7) (Final Office Action, page 42);

6) When taken as a whole, Martin teaches "actors" being defined and rules associated with the tasks of the participants on page 144 (Final Office Action, page 43);

7) participants are objects to which the integrity rules of Martin may be applied (Final Office Action, page 44);

8) In re Graves supports grounds of obvious rejections if an artisan of ordinary skill would have the knowledge at the time of invention on how to implement the basics of the tools of the trade as taught by Martin (Final Office Action, page 47);

9) The anatomy of a rule is demonstrated with common programming constructs (Final Office Action, page 48);

10) The Martin reference teaches the checking of integrity and Code Complete reference teaches common sense is to be used. The IBM dictionary underscores the industry use of the terms which took years to culminate into a denotation and published in a computer dictionary (Final Office Action, pages 49-50); and

11) One of ordinary skill in the art should know that the calling of a function transfers control and the completion returns control. Furthermore, the ability to check a return code in programming is grossly old and well known (Final Office Action, page 51).

The above responses by the Examiner may be summarized in one basic misplaced concept – the Examiner believes that any general teaching renders obvious all specific features the Examiner may argue are related, no matter how remotely related they may be. As a result, simply because Martin generally teaches rules and integrity checking rules, Martin now miraculously teaches all of the specific features set forth in Appellants' claims. This is clearly an incorrect approach to examination of patent claims. If such a notion were valid, all computer program claims would be obvious simply by virtue of the fact that there are textbooks that teach the basic constructs of programming for performing functions which could possibly be combined to achieve a claimed computer program, even though such combinations are not taught or suggested by the textbook. The Examiner seems to believe that simply because basic constructs

could possibly be combined to arrive at a claimed feature, that the mere possibility in itself is the suggestion to perform the combination. Thus, using the Examiner's reasoning, simply because Martin teaches object oriented programming, any program that operates in an object oriented environment is now obvious in view of Martin simply because the basic constructs in Martin could possibly be combined to generate any object oriented program, even though such combinations are not taught or suggested by Martin. This is clearly not the case.

All of the Examiner's responses to Appellants arguments are rooted in this erroneous concept. The Examiner as much as explicitly states this when the Examiner refers to *in re Graves* (see response # 8 above) as allegedly supporting the proposition that the grounds of an obvious rejection are proper if an artisan or ordinary skill would have the knowledge at the time of invention on how to implement the basics of the tools of the trade as taught by Martin (Final Office Action, page 47). The Examiner is misapplying *in re Graves*. While one of ordinary skill in the art might arguably have the knowledge provided in Martin with regard to general concepts regarding object oriented programming, rules, and integrity checking rules, the mere knowledge is not sufficient without a suggestion or incentive to modify or combine this knowledge in such a way as to arrive at Appellants' claimed invention. What would prompt one of ordinary skill in the art to combine and modify such knowledge to arrive at the specific features in Appellants' claimed invention? There is nothing in Martin that explicitly teaches or suggests the features of the present invention, as discussed above.

To reduce the Examiner's position to its basic premise, the Examiner seems to believe that Appellants are merely claiming integrity checking rules in general and thus, the general teaching of integrity checking rules in Martin is used as a basis for rejecting all of claims 1-29. However, Appellants are not merely claiming any integrity checking rules but rather, as recited in claim 1, for example, a very specific mechanism for running rules for each of a plurality of participants in a unit of work in order to verify the integrity of an application state, in response to a determination that a unit of work is to be completed, at a point in the unit or work where application state integrity is to be verified. Martin does not teach such a mechanism. All that Martin teaches is that integrity checking rules may be used and attached to diagrams. There is nothing in Martin that teaches determining a point in a unit of work where the integrity of an application state is to be verified, determining if a unit of work is to be completed, obtaining rules for each of a plurality of participants in the unit of work in response to a determination that

the unit of work is to be completed, and running the rules for each of the participants to verify the application state. Rather than finding a reference that explicitly teaches or suggests these specific features of the claimed invention, the Examiner insists on citing a general text book and making tenuous associations of teachings in Martin with claimed features when those teachings in Martin have nothing to do with the features of the pending claims.

With regard to the Examiner's response that Martin teaches object oriented CASE tools and common programming constructs (see response #1 above), Appellants are not in disagreement. Appellants' position is that even though Martin teaches such common programming constructs, these are not the same as the features recited in the claims. These common programming constructs do not teach or suggest identifying a point in a unit of work where application state integrity is to be verified, obtaining rules for a plurality of participants in the unit of work in response to a determination that the unit of work is to be completed, or running the rules for each of the participants to verify integrity of an application state. The mere general teaching of integrity checking rules does not rise to the level required to obviate these features.

Regarding the Examiner's response that Appellants allegedly admit that the Martin reference teaches "units of work" (see response #2 above), Appellants in no way admitted that Martin teaches units of work. Appellants' acknowledgment that Martin teaches that business policies may be implemented as rules is in no way an admission that Martin teaches units of work. Business policies are not units of work that are to be performed and have a plurality of participants that operate to complete the unit of work. Rather, a business policy merely states the general decision making of an enterprise solution. There is no correlation between business policies and units of work in the Martin reference. Thus, Appellants maintain that Martin does not teach units of work, despite the Examiner's allegation.

With regard to the Examiner's response that the general teaching of integrity rules and attaching integrity rules to diagrams renders obvious the feature of identifying a point in a unit of work where state integrity of an application is to be checked (see response # 3 above), Appellants respectfully disagree. Again the Examiner is taking general teachings and warping them to fit into the mold of Appellants' claims. There is nothing in a general teaching of attaching integrity rules to diagrams that can reasonably be interpreted as teaching identifying a point in a unit of work where application state integrity is to be checked. Where is the unit of work in a diagram?

Where are the plurality of participants in the unit of work illustrated in a diagram? Where is the obtaining of rules for the plurality of participants in the unit of work shown in the diagram? Where is the running of rules for the plurality of participants in order to verify application state shown in the diagram?

Regarding the Examiner's response that any judgment of obviousness is necessarily a reconstruction based on hindsight reasoning (see response #4 above), Appellants agree. However, it is Appellants' position that the Examiner has not made a reconstruction based solely on the knowledge which was within the level of ordinary skill in the art at the time the claimed invention was made. The Examiner has based the rejection of claims 1-29 completely on knowledge gleaned only from Appellants' disclosure. Other than the general teaching of integrity checking rules, there is nothing in Martin that remotely resembles anything in the present claims. Thus, the Examiner, rather than finding a reference that teaches or suggests the features of the present claims, takes these general teachings of Martin and attempts to warp their interpretation to fit the mold defined by Appellants' claims. Such a process is based completely on hindsight. Therefore, while a judgment of obviousness must have some measure of hindsight, the finding of obviousness in the present case is completely based on hindsight. *In re McLaughlin* does not support the basis of the present rejection.

With regard to the Examiner's response that the Martin reference, when taken as a whole, teaches object oriented CASE tools for enterprise solutions and enterprises have many participants (see response #5 above), Appellants respectfully disagree that such a teaching renders obvious the features of the claimed invention. While Martin may teach objects and interaction of objects generally, there is nothing in Martin that teaches a unit of work having a plurality of participants and the identification of a point in the unit of work where application state integrity is to be checked. Moreover, there is nothing in Martin that teaches the obtaining of rules associated with each participant in a unit of work when it is determined that the unit of work is to be completed. Further, there is nothing in Martin that teaches running these rules for each of the participants to verify application state in accordance with the plurality of participants. The mere teaching of objects and objects being able to interact with one another does not rise to the level to obviate these features of the present invention.

With regard to the Examiner's response that Martin, when taken as a whole, teaches "actors" being defined and rules associated with the tasks of the participants on page 144 (see

response #6 above), Appellants respectfully disagree. Page 144 of Martin only teaches examples of business rules associated with diagrams for object-oriented analysis. There is nothing on page 144 or any other section of Martin that teaches or suggests a unit of work having a plurality of participants, identifying a point in the unit of work where application state is to be checked, obtaining rules for the participants in response to a determination that a unit of work is to be completed, or running the rules for the participants to verify application state.

Regarding the Examiner's response that participants are objects to which the integrity rules of Martin may be applied (see response #7 above), while the participants in a unit of work according to the present invention may be objects (see for example, pending claim 5), Appellants disagree that the participants as defined in the present claims are objects to which integrity rules of Martin may be applied. First, even if integrity rules of Martin may be "applied" to objects, this still does not teach or suggest identifying a point in a unit of work where application state integrity is to be checked. Moreover, there is nothing in Martin that teaches that each of the participants in a unit of work have their own rules that are run to verify application state. Likewise, there is nothing in Martin that teaches the need to obtain such rules when it is determined that a unit of work is to be completed.

Regarding the Examiner's application of *in re Graves* (see response #8 above), this misapplication of case law is addressed above. With regard to the Examiner's response that the anatomy of a rule is demonstrated with common programming constructs (see response #9 above), the Examiner's response does not address Appellants' argument. That is, simply teaching that a rule is comprised of common programming constructs has nothing to do with determining whether a unit of work is to be completed and obtaining rules for each of the participants in the unit of work in response to determining that the unit of work is to be completed.

With regard to the Examiner's response that the Martin reference teaches the checking of integrity and Code Complete reference teaches common sense is to be used; and the IBM dictionary underscores the industry use of the terms which took years to culminate into a denotation and published in a computer dictionary (see response #10 above), Appellants respectfully submit that "common sense" and well known terms do not cure the deficiencies in Martin detailed repeatedly above. The use of "common sense" does not provide any teaching or suggestion to modify Martin to include an identification of a point in a unit of work where

application state integrity is to be checked, obtaining rules for a plurality of participants in a unit of work in response to a determination that the unit of work is to be completed, or running the rules for each of the participants to verify the application state. Well-known terms, likewise, do not provide any of these teachings.

Regarding the Examiner's response that one of ordinary skill in the art should know that the calling of a function transfers control and the completion returns control; and furthermore, the ability to check a return code in programming is grossly old and well known (see response #11 above), Appellants respectfully submit that this has nothing to do with determining whether a unit of work is to be completed. Nowhere in Martin is it taught to determine whether a unit of work is to be completed. The allegation that checking return codes is the same as this claimed feature is simply wrong. Checking return codes has nothing to do with determining if a unit of work is to be completed and, in response to determining that the unit of work is to be completed, obtaining rules associated with each of the participants in the unit of work.

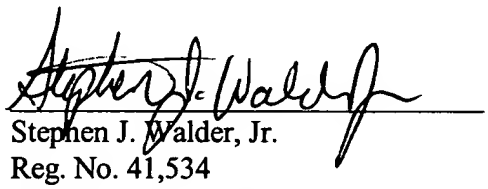
III. Summary

In summary, the Examiner has merely taken general teachings regarding basic concepts of object oriented programming as taught by Martin and has stretched and warped their interpretation in an attempt to fit the specific features of the present invention. There is nothing in Martin, Code Complete, or the IBM Computer Dictionary that has anything to do with the present invention as recited in claims 1-29 other than a general teaching of integrity checking rules. There is nothing in any of these references that teaches or suggests a unit of work having a plurality of participants, identifying a point in the unit of work where application state integrity is to be checked, obtaining rules for each of the participants in the unit of work in response to a determination that the unit of work is to be completed, and running the rules for each of the participants in order to verify application state integrity.

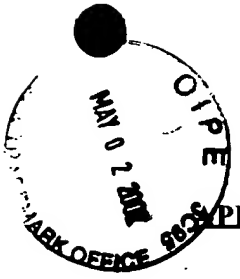
IV. Conclusion

In view of the above, Appellants respectfully submit that all of claims 1-29 define over the prior art of record, Martin, Code Complete, and the IBM Computer Dictionary. Appellants therefore, request that the Board of Patent Appeals and Interferences overturn the rejection of claims 1-29 under 35 U.S.C. § 103(a).

Respectfully submitted,



Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
PO Box 802334
Dallas, TX 75380
(972) 367-2001



APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

1. (Once Amended) A method for performing general integrity checks using rules in an application running on a data processing system comprising:

identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;

responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and

responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants.

2. (Unchanged) The method of claim 1, further comprising:

responsive to a negative result obtained by running the rules, aborting the unit of work.

3. (Unchanged) The method of claim 1, further comprising:

responsive to a positive result obtained by running the rules, committing the unit of work.

4. (Unchanged) The method of claim 1, wherein each participant is associated with a name and wherein the step of obtaining rules associated with each participant in the unit of work comprises obtaining rules based on the name associated with the participant.

5. (Unchanged) The method of claim 4, wherein the plurality of participants are a plurality of objects and wherein the name associated with an object within the plurality of objects is the class name of a participating object.

6. (Unchanged) The method of claim 1, wherein each participant is associated with a name, wherein the unit of work is associated with a type, and wherein the step of obtaining rules associated with each participant in the unit of work comprises obtaining rules based on the name associated with the participant and the type associated with the unit of work.

7. (Once Amended) The method of claim 1, wherein at least zero integrity checking rules are associated with each participant within the plurality of participants.

8. (Unchanged) A method in a data processing system for performing general integrity checks using rules, the method comprising:

detecting a commit for a unit of work;

identifying participants in the unit of work in response to detecting the commit for the unit of work;

determining whether rules are present for the participants in the unit of work;

running the rules for participants identified as having at least one rule;

determining whether a violation of an integrity rule within the rules identified for any participant has occurred; and

committing the unit of work depending on the results of running the rules.

9. (Unchanged) The method of claim 8 further comprising:

aborting completion of processing by the unit of work in response to a determination that a violation of a rule has occurred; and

committing completion of processing by the unit of work in response to a determination that no violation of a rule has occurred.

10. (Unchanged) The method of claim 8, wherein each participant has zero or more rules associated therewith.

11. (Unchanged) The method of claim 8, wherein each rule associated with a unit of work has available for use each participant within the unit of work.

12. (Unchanged) An enterprise application for use in a computer, the enterprise application comprising:

a unit of work, wherein the unit of work accumulates participants that affect a state of the enterprise application;

a plurality of business rules, wherein the plurality of rules are used to verify the integrity of the application state; and

a unit of work control point, wherein the unit of work control point locates applicable rules for participants in response to an activation of the unit of work to complete processing of the unit of work.

13. (Unchanged) The enterprise application of claim 12, wherein the activation of the unit of work control point for the unit of work is initiated by a commit instruction to the unit of work.

14. (Unchanged) The enterprise application of claim 12, wherein the control point identifies applicable rules for all of the participants in the unit of work.

15. (Unchanged) The enterprise application of claim 12, wherein the control point applies applicable rules to a portion of the participants in the unit of work.

16. (Unchanged) The enterprise application of claim 12, wherein the applicable rules are identified based on a name associated with the participant.

17. (Unchanged) The enterprise application of claim 12, the participant is an object and wherein the name is the class name of the participating object.

18. (Unchanged) The enterprise application of claim 17, wherein the unit of work is associated with a type and wherein the applicable rules also are identified based on the type associated with the unit of work.

19. (Unchanged) A data processing system for performing general integrity checks using rules in an application running on a data processing system comprising:

identifying means for identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;

first obtaining means, responsive to determining that the unit of work is to be completed, for obtaining rules associated with each participant in the unit of work; and

second obtaining means, responsive to obtaining the rules, for running the rules obtained for each of the participants to verify the integrity of the system, according to the plurality of participants.

20. (Unchanged) The data processing system of claim 19, further comprising:

aborting means, responsive to a negative result obtained by running the rules, for aborting the unit of work.

21. (Unchanged) The data processing system of claim 19, further comprising:

committing means, responsive to a positive result obtained by running the rules, for committing the unit of work.

22. (Unchanged) A data processing system for performing general integrity checks using rules, the data processing system comprising:

detecting means for detecting a commit for a unit of work;

identifying means for identifying participants in the unit of work in response to detecting the commit for the unit of work;

first determining means for determining whether rules are present for the participants in the unit of work;

running means for running the rules for participants identified as having at least one rule;

second determining means for determining whether a violation of an integrity rule within

the rules identified for any participant has occurred; and

committing means for committing the unit of work depending on the results of running the rules.

23. (Unchanged) The data processing system of claim 22 further comprising:

aborting means for aborting completion of processing by the unit of work in response to a determination that a violation of a rule has occurred; and

committing means for committing completion of processing by the unit of work in response to a determination that no violation of a rule has occurred.

24. (Unchanged) The data processing system of claim 22, wherein each participant has zero or more rules associated therewith.

25. (Unchanged) A computer program product for performing general integrity checks using rules in an application running on a computer program product comprising:

first instructions for identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;

second instructions for responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and

third instructions for responsive to obtaining the rules, running the rules obtained for each of the participants to verifying the integrity of the system, according to the plurality of participants.

26. (Unchanged) The computer program product of claim 25, further comprising:

first instructions for responsive to a negative result obtained by running the rules, aborting the unit of work.

27. (Unchanged) The method of claim 25, further comprising:

first instructions for responsive to a positive result obtained by running the rules, committing the unit of work.

28. (Unchanged) A computer program product in a data processing system for performing general integrity checks using rules, the computer program product comprising:

first instructions for detecting a commit for a unit of work;

second instructions for identifying participants in the unit of work in response to detecting the commit for the unit of work;

third instructions for determining whether rules are present for the participants in the unit of work;

fourth instructions for running the rules for participants identified as having at least one rule;

fifth instructions for determining whether a violation of an integrity rule within the rules identified for any participant has occurred; and

sixth instructions for committing the unit of work depending on the results of running the rules.

29. (Unchanged) The computer program product of claim 28 further comprising:

first instructions for aborting completion of processing by the unit of work in response to a determination that a violation of a rule has occurred; and

second instructions for committing completion of processing by the unit of work in response to a determination that no violation of a rule has occurred.